Designing Run-time Self-Evolution and Self-Adaptation for Smart Software Systems

Researchers

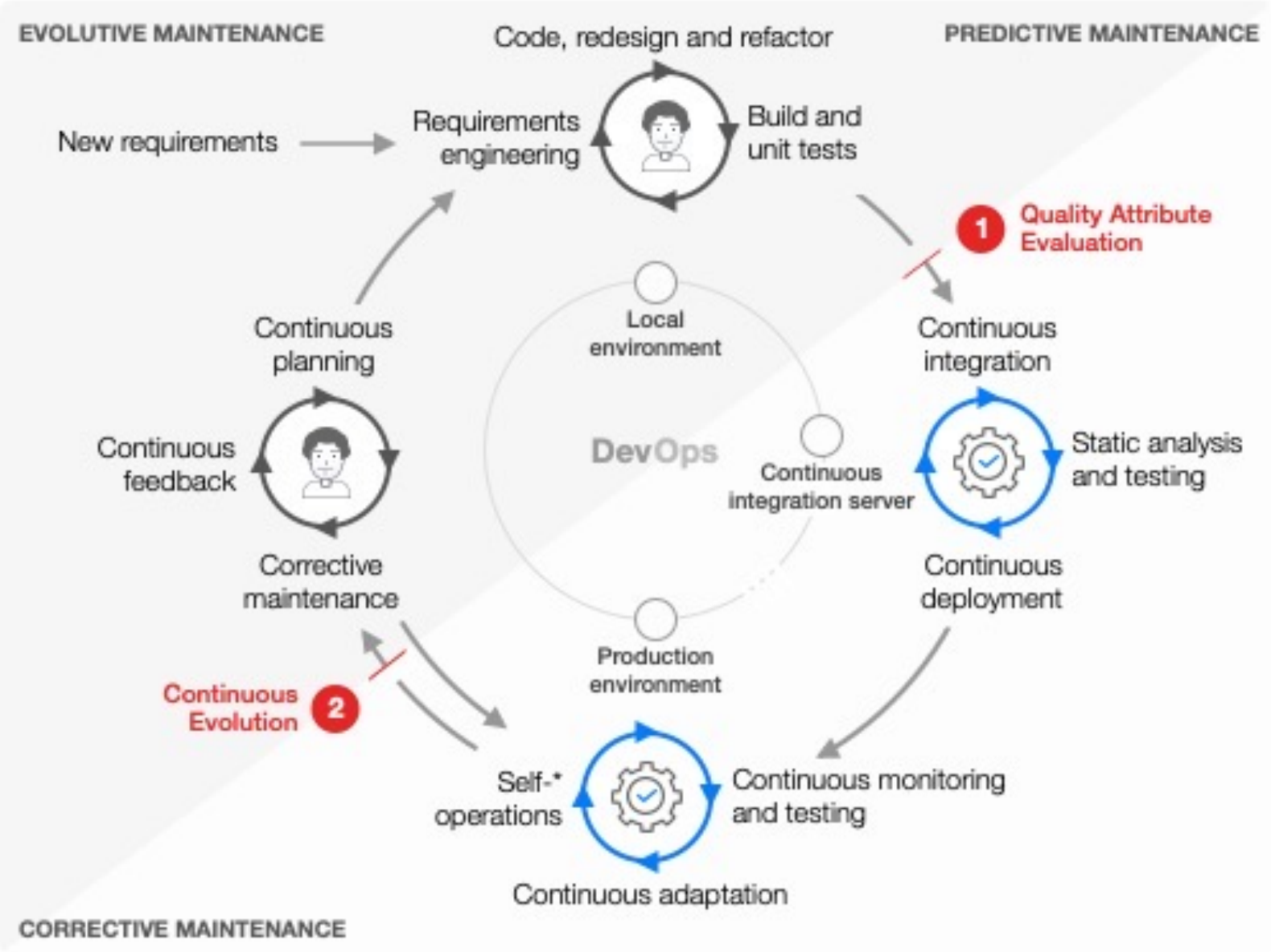
Miguel Jiménez, Felipe Rivera, Norha M. Villegas, Gabriel Tamura, Hausi Müller

IBM Collaborators

Joe Wigglesworth, Ian Watts, Hugh Hockett

April 22, 2021
Victoria, Canada

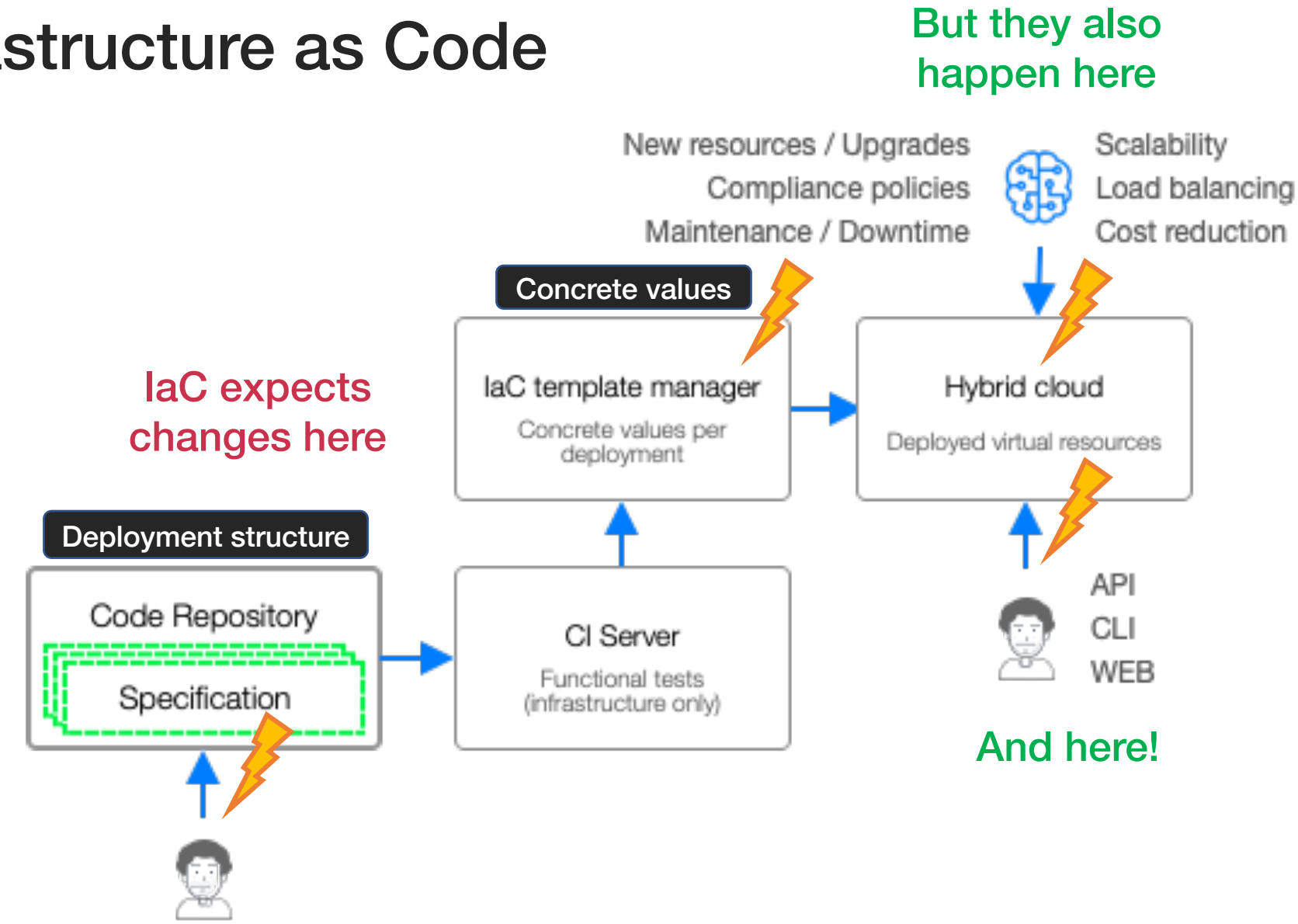
Discontinuities in the Delivery Process



- Short-term evolution is well understood
- Long-term evolution remains largely unexplored
- QA evaluation is not well integrated, e.g., software patterns
- AIOps is mostly about root cause analysis

Example #1: Infrastructure as Code

- This is what a **continuous** process looks like: DevOps engineers working on both sides of the software lifecycle.
- Systematic approaches to maintain the correspondence between design and code are rarely used in practice*
- Limited integration restricts autonomic managers from doing more impactful contributions



* Nugroho, Ariadi, and Michel RV Chaudron. "**A survey of the practice of design--code correspondence amongst professional software engineers.**" 2007.

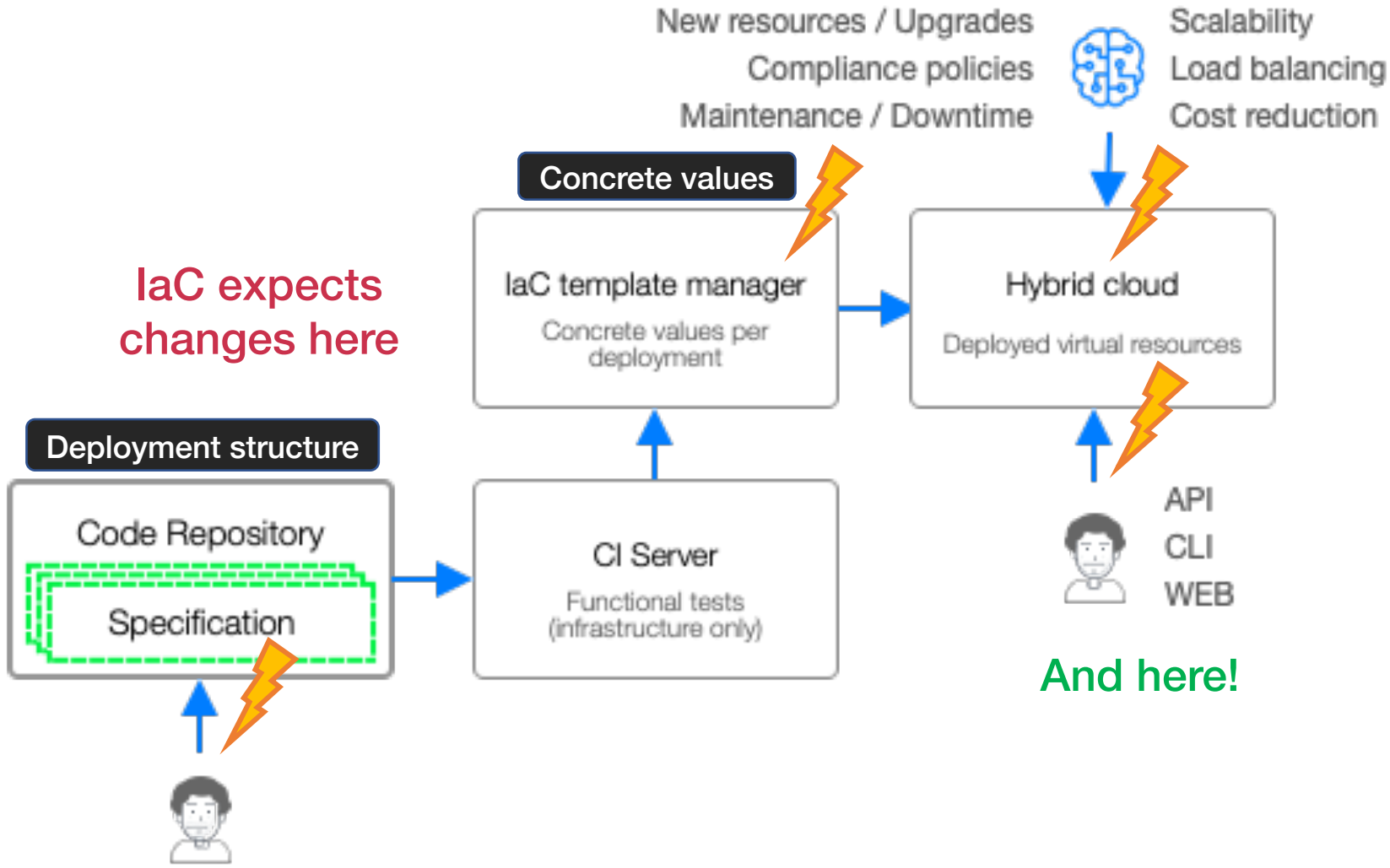
Example #1: Infrastructure as Code

Adoption of IaC

- Thousands of resources
- Manual creation is error-prone
- Costly process

Evolution of specifications

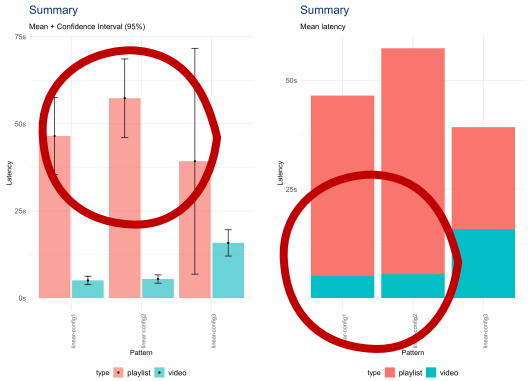
- Discard existing automation
- Inconsistencies lead to technical debt



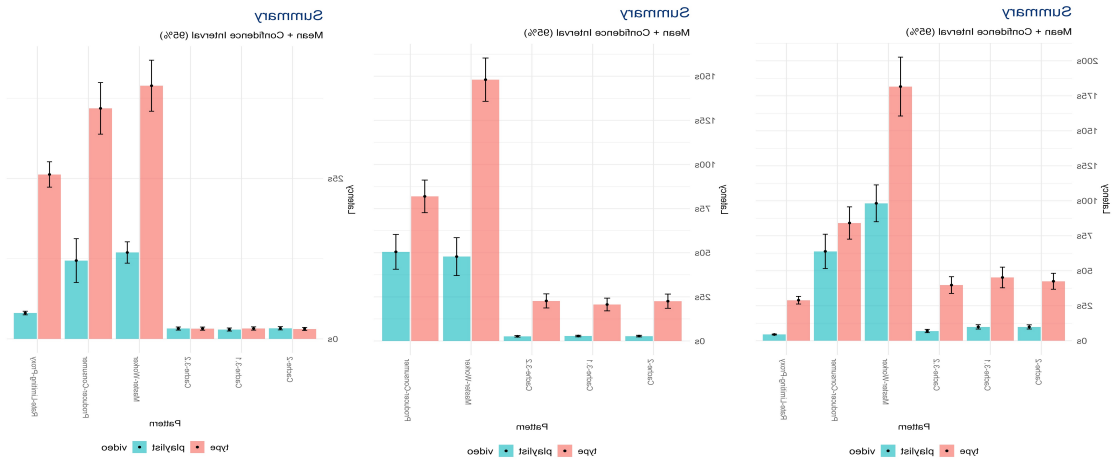
* Nugroho, Ariadi, and Michel RV Chaudron. "**A survey of the practice of design--code correspondence amongst professional software engineers.**" 2007.

Example #2: Software patterns evaluation

Automatic cluster configuration and tuning



Software architecture as a function of the environment



Constant **Linear** **Spike**

- Quantitative evaluation of software patterns (including deployment)
- Are descriptions enough in Software Engineering?
- How does QA evaluation fit into continuous delivery?
- Lack of integration restricts more advanced evolution at run-time

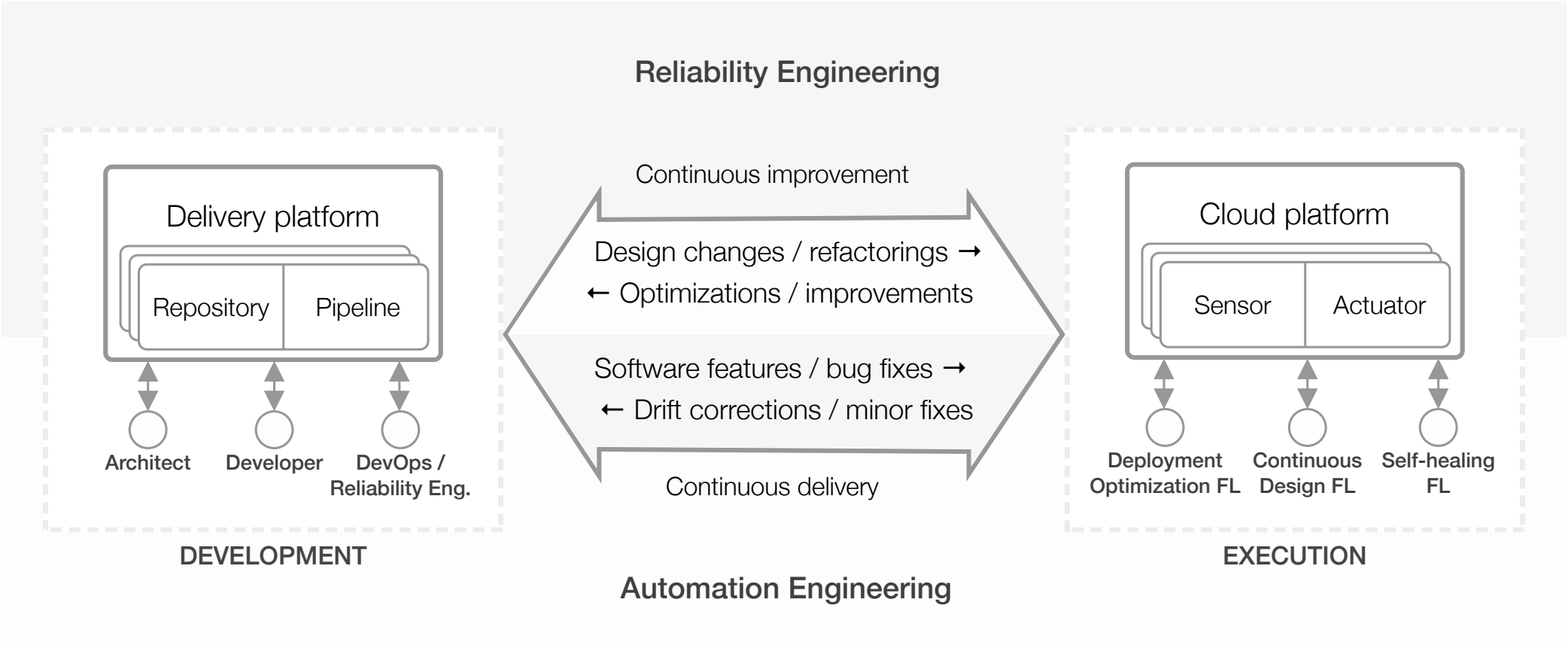
Deployment specification/management challenges

1. Notation and tool support for linking design-time and run-time deployment concepts

2. Tool support for the evolution of deployment specifications and configuration management at runtime

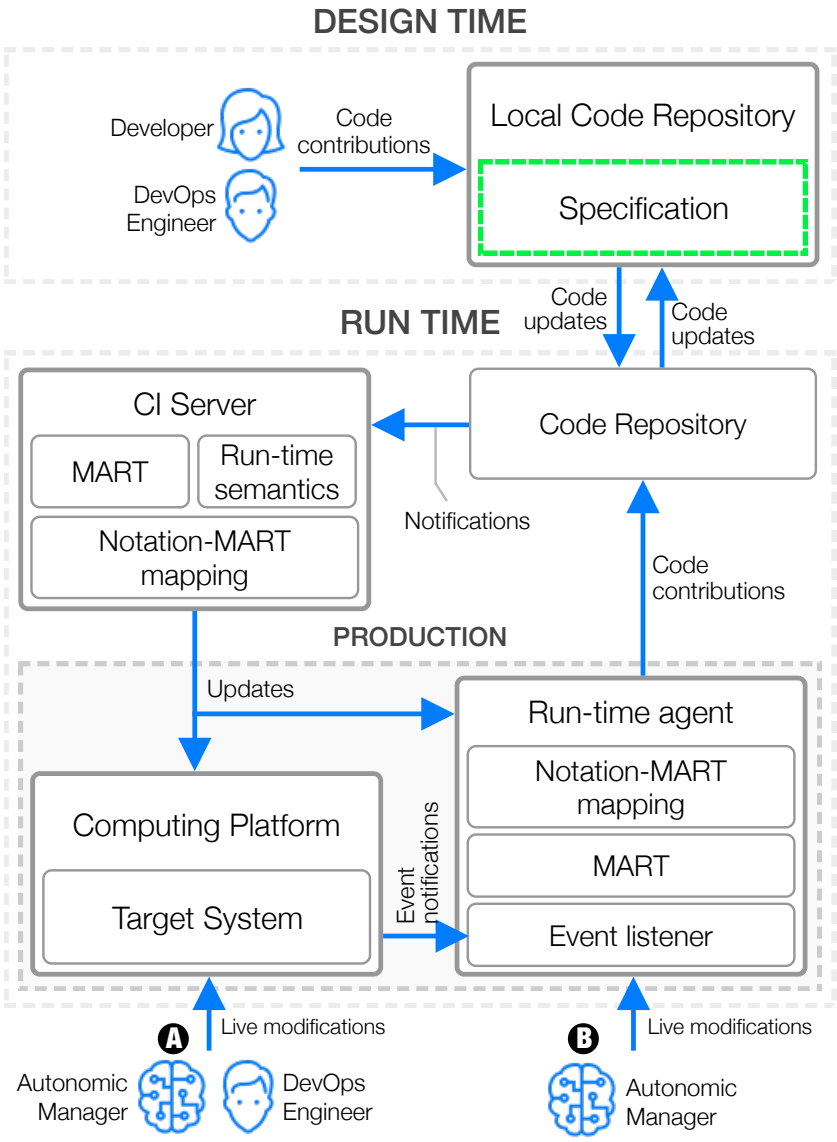
Two-way continuous delivery

- Automation vs reliability engineering
- Continuous delivery vs improvement
- Independent, co-existing feedback loops



Two-way continuous integration

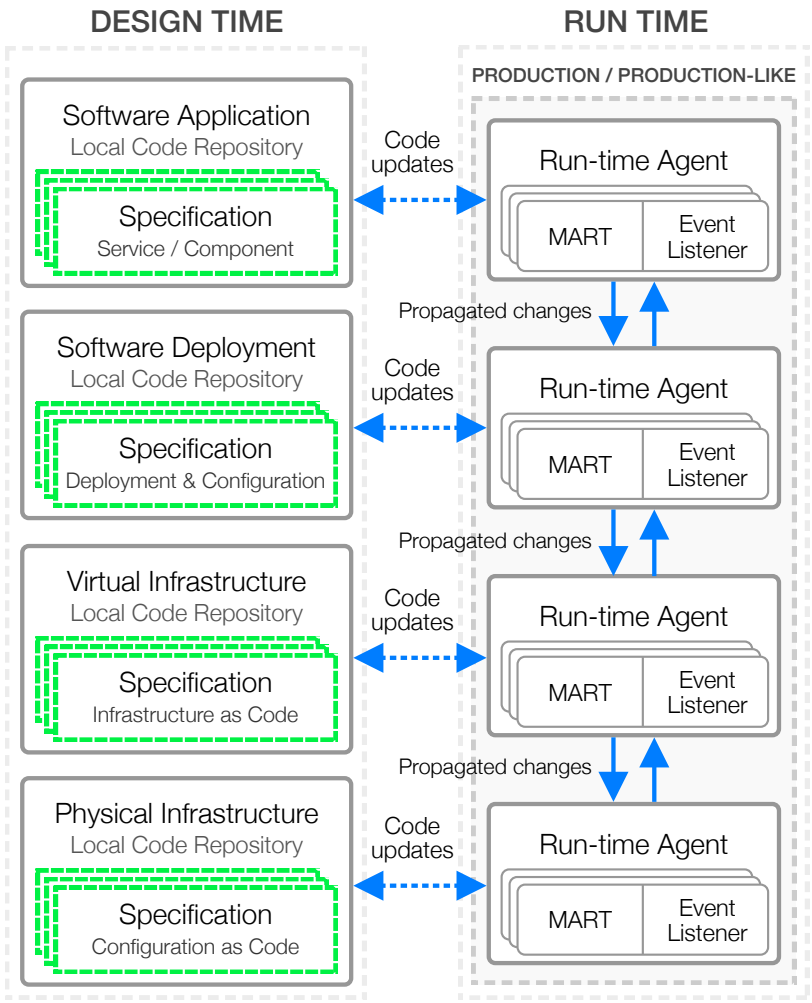
- Run-time models map live changes with code updates
- Baseline for long-term evolution

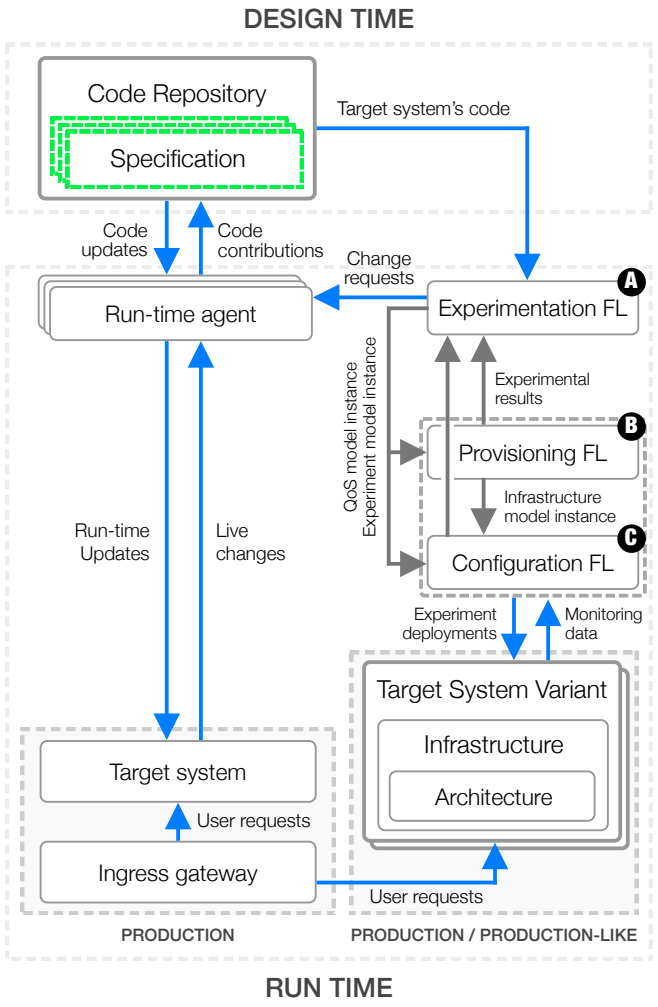


Miguel Jiménez *et al*. **DevOps Round-Trip Engineering: Traceability from Dev to Ops and Back Again.** 2019

Quality-driven experimentation

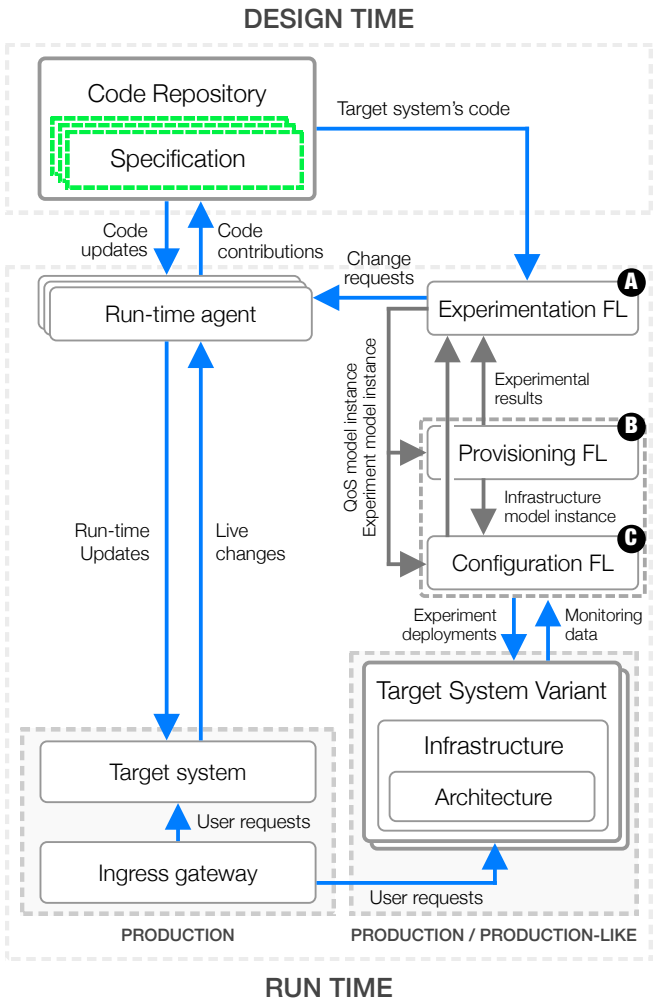
- Mesh of run-time models to represent cross-cutting concerns
- Experimentation system to tune the infrastructure and the architecture at run-time
- Integration with development is still missing (QA evaluation)
- It's closer to our long-term vision but lacks a more generic definition

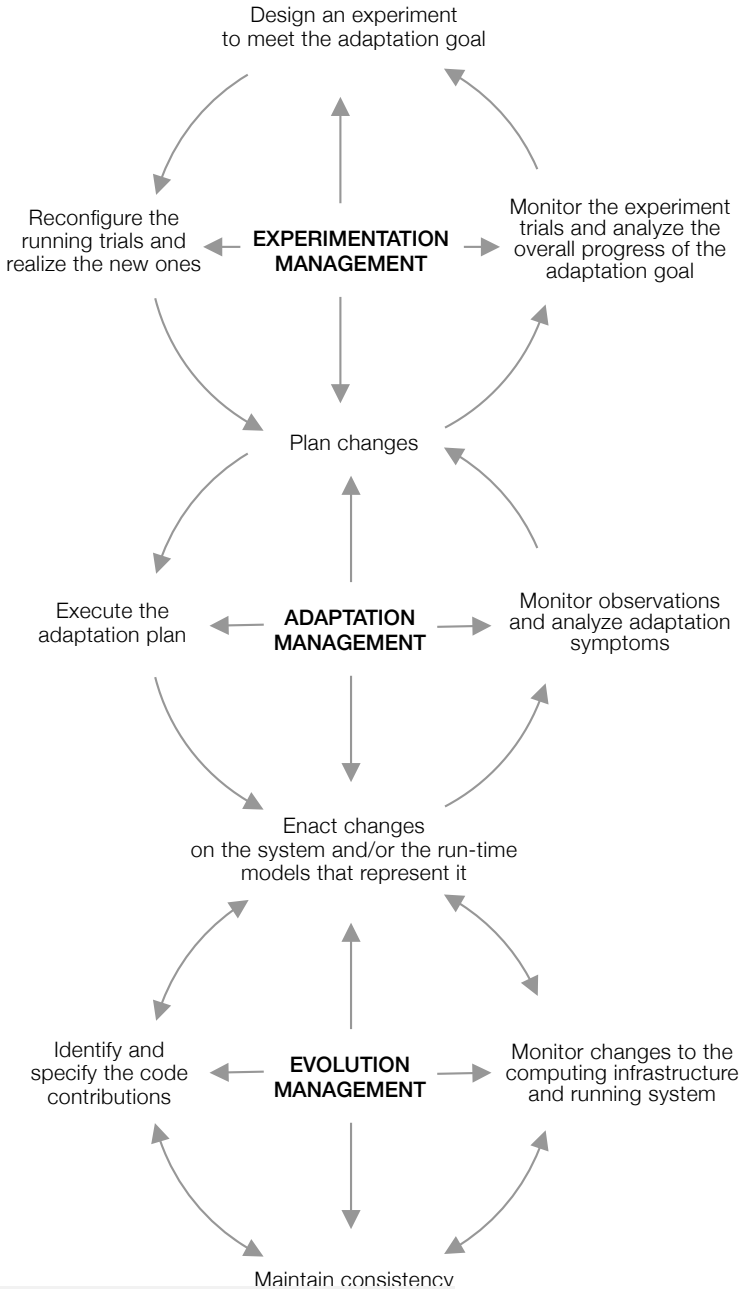




Quality-driven experimentation

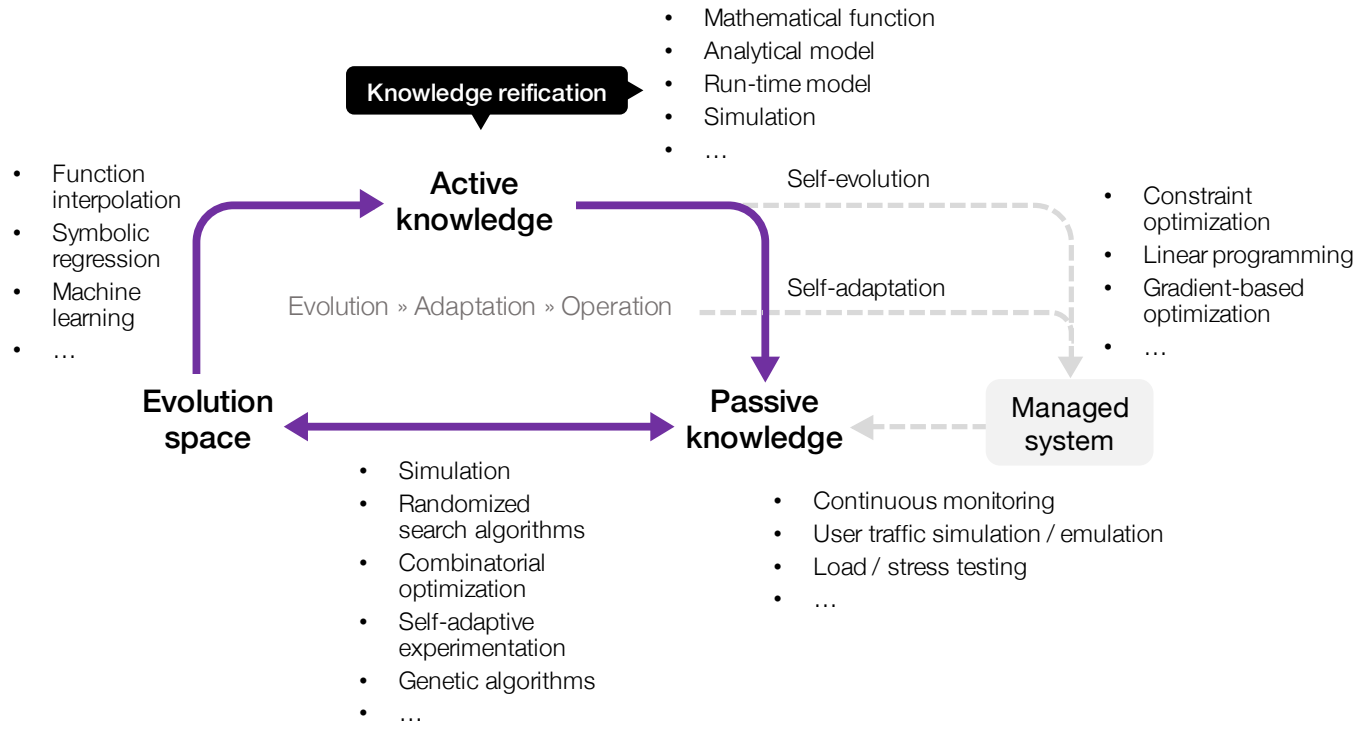
- Mesh of run-time models to represent cross-cutting concerns
- Experimentation system to tune the infrastructure and the architecture at run-time
- Integration with development is still missing (QA evaluation)
- It's closer to our long-term vision but lacks a more generic definition

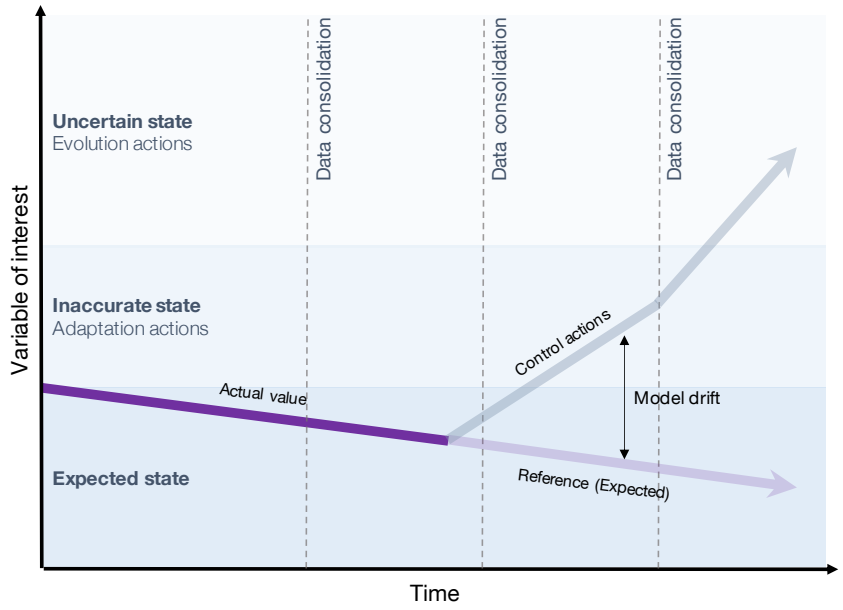




Toward an evolution process

- Relationship between evolution, adaptation and operation
- Passive vs active knowledge
- Techniques for knowledge reification

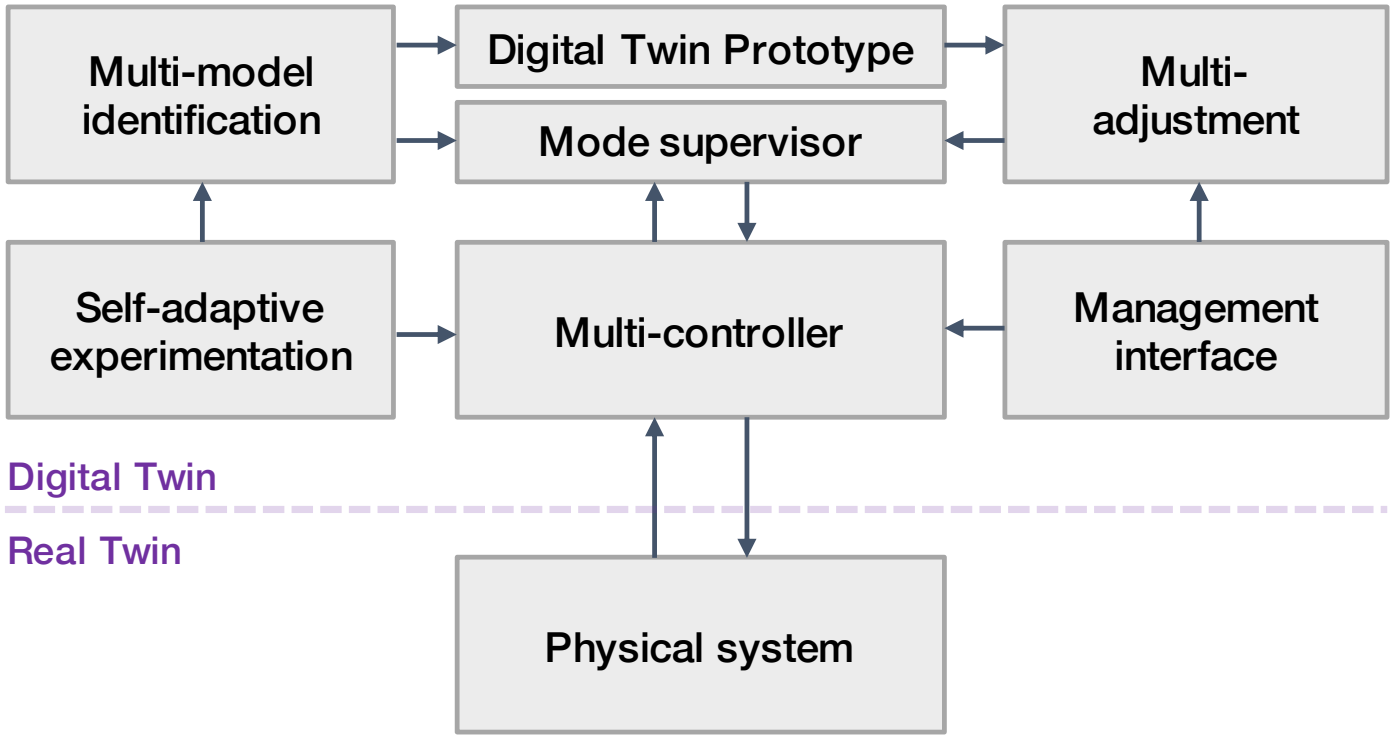




L.F. Rivera, Miguel Jiménez et al. **Designing Run-time Evolution for Dependable and Resilient Cyber-Physical Systems Using Digital Twins.** 2021. In press.

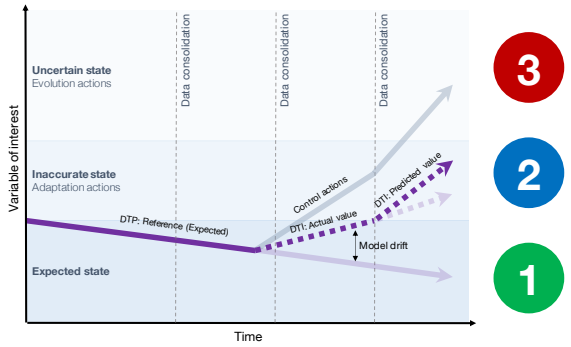
A reference architecture

- Dependable autonomy
- Operational resiliency
- Evolution vs adaptation duality (long- vs short-term)
- Originally proposed for cyber-physical systems



L.F. Rivera, Miguel Jiménez et al. **Designing Run-time Evolution for Dependable and Resilient Cyber-Physical Systems Using Digital Twins.** 2021. In press.

A run-time evolution process (work in progress)



3

2

1

New release / code update
Dev → Ops

RUN-TIME ASSURANCE

Adaptation space exploration

Control

2

Passive knowledge

MODEL IDENTIFICATION

Active knowledge

Long-term updates

Short-term updates

Data collection

Evolution space exploration

Knowledge reification

3

Persistent update

Operation

1

Run-time V&V

DevOps

Development ⟷ Operations

Cyber-Physical Systems

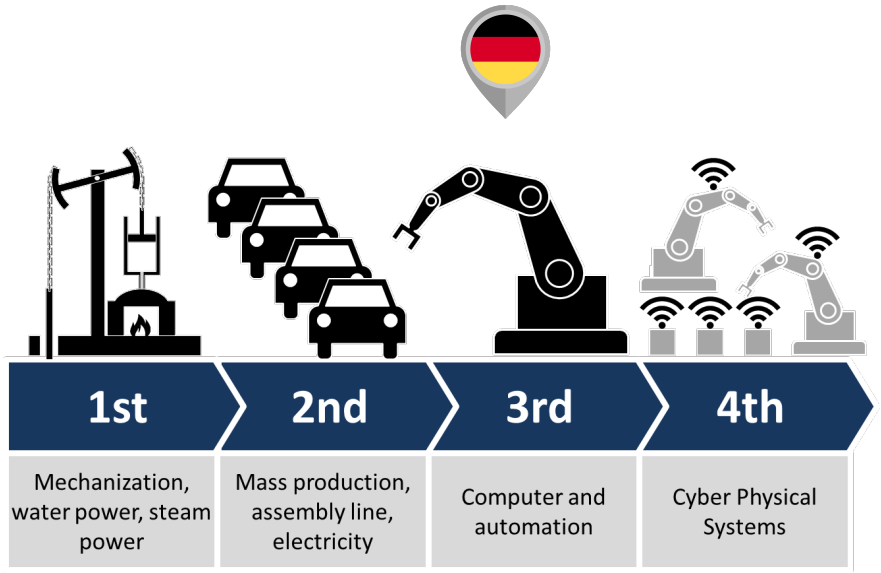
Physical system ⟷ Virtual system

Real twin ⟷ **Digital twin**

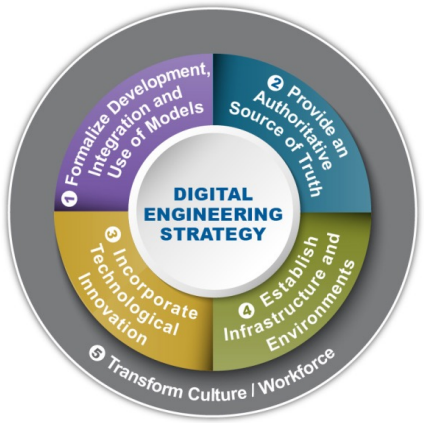
Dev-Ops duality beyond Software Engineering

- Impact in both industry and society

Industry

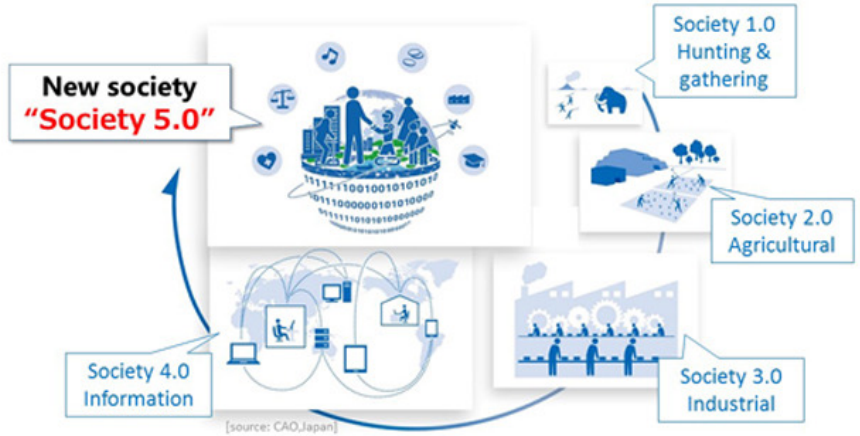


Industry 4.0 [1]
Image adopted from Roser [2].



Digital Engineering [3]
Image adopted from USA DoD [3].

Society



Society 5.0 [4]
Image adopted from CAO Japan [5].

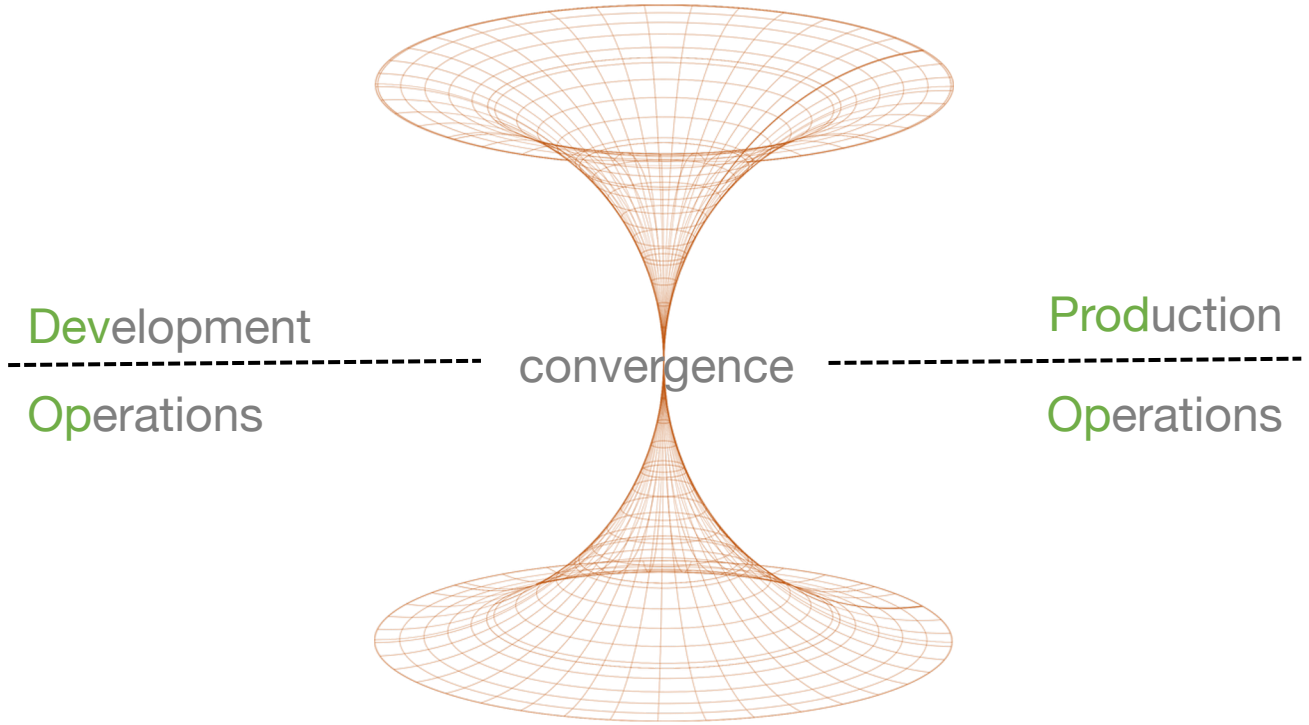
Robust Software Systems

[1] H. Lasi, P. Fettke, H.G. Kemper, T. Feld, and M. Hoffmann. Industry 4.0. Business & Information Systems Engineering. 2014.
[2] C. Roser. Illustration of Industry 4.0. www.allaboutlean.com. 2016.
[3] USA Department of Defense (Dod). Department of Defense - Digital Engineering Strategy. 2018.
[4] Cabinet Office. The 5th Science and Technology Basic Plan. Government of Japan. 2016.
[5] Cabinet Office. Society 5.0. www8.cao.go.jp/cstp/english/society5_0/index.html. 2020.

*Icons made by Freepik from www.flaticon.com.

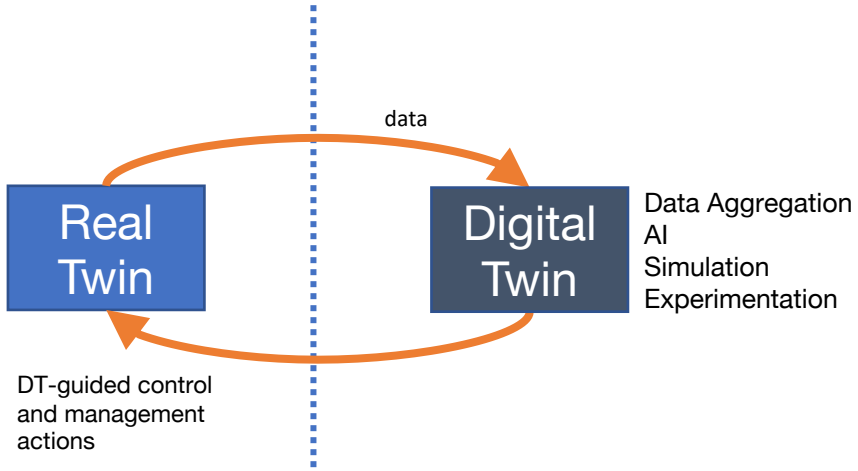
Dev-Ops duality beyond Software Engineering: Convergence

- Convergence between production and operations requires the digitalization of physical assets
- Digital-Physical convergence



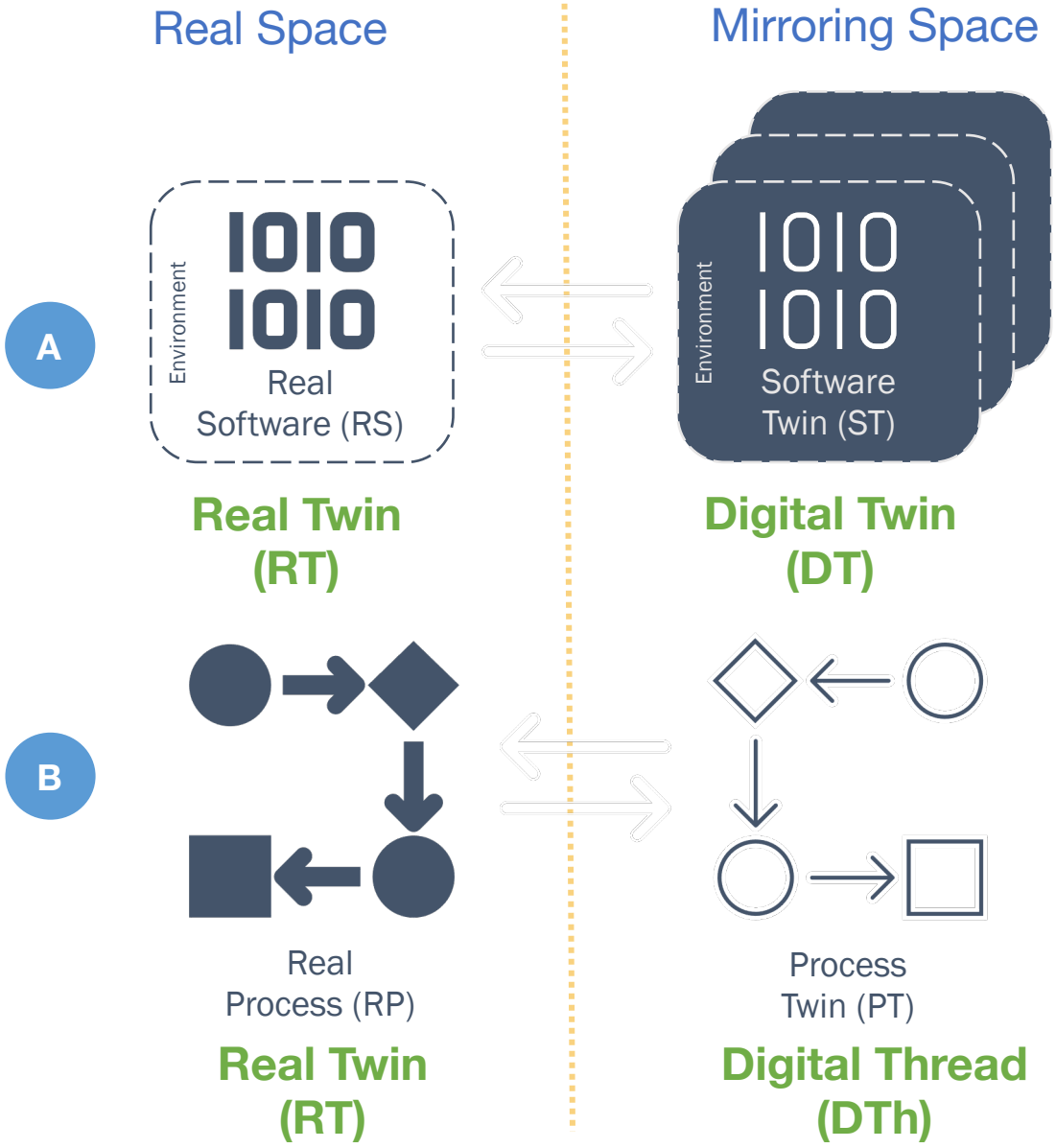
Digital Twin (DT)

- A virtual representation of a physical or conceptual real-world entity (RT) throughout its entire lifecycle.

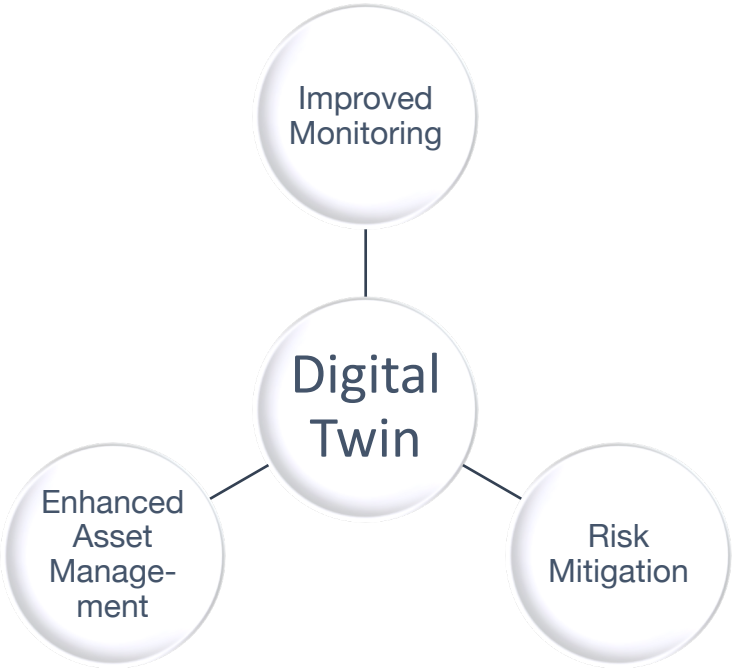


Representation | Structure | Behaviour

Models | Sensors | Data | Life cycle

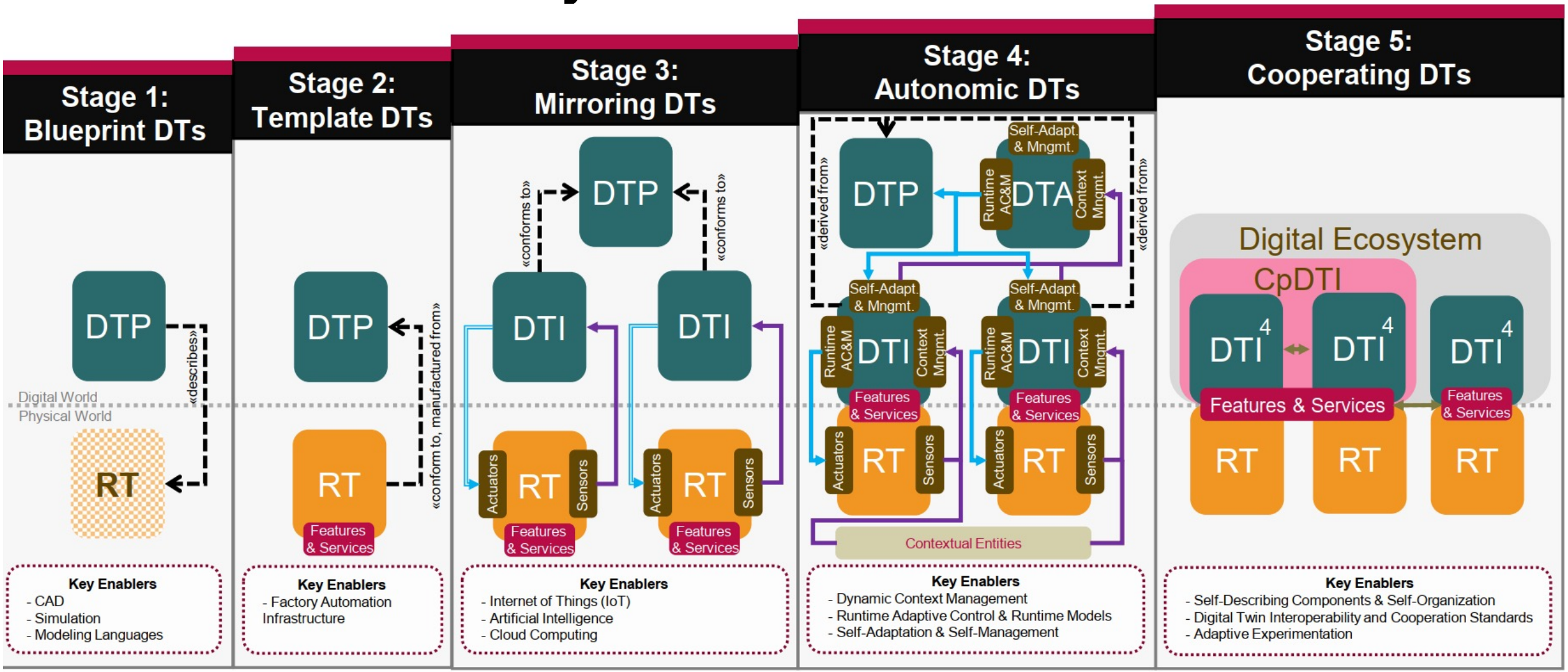


DT benefits and trend

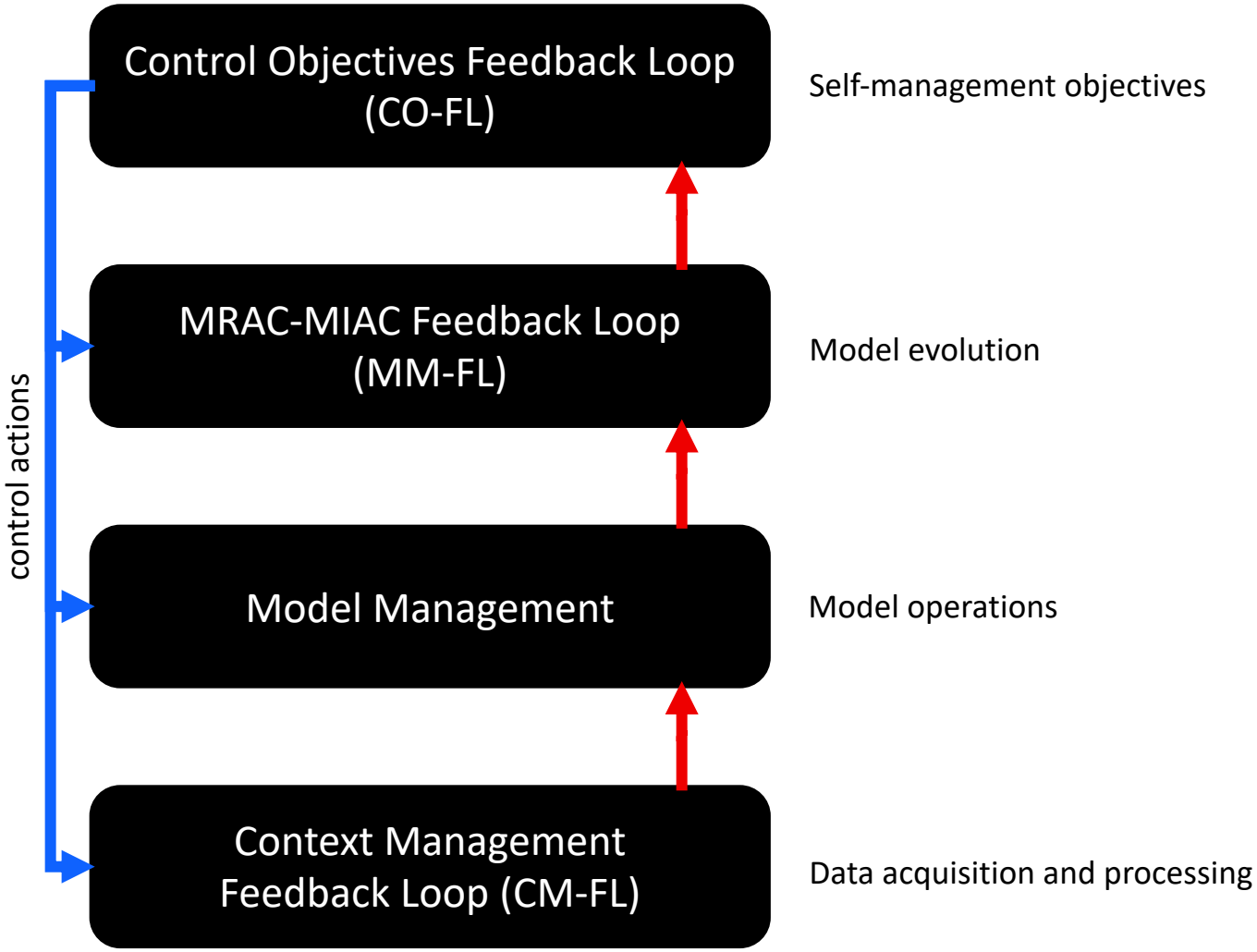


Ongoing CAS Project

DT Evolution: Maturity Model



The GEMINIS Reference Model




2020 IEEE/ACM 42nd International Conference on Software Engineering Workshops (ICSEW)

On the Engineering of IoT-Intensive Digital Twin Software Systems

Luis F. Rivera, rivera@uvic.ca, University of Victoria, Victoria, British Columbia, Canada
Hausi A. Müller, hausi@uvic.ca, University of Victoria, Victoria, British Columbia, Canada
Norha M. Villegas, nvillega@icesi.edu.co, Universidad Icesi, Cali, Valle del Cauca, Colombia
Gabriel Tamura, gtamura@icesi.edu.co, Universidad Icesi, Cali, Valle del Cauca, Colombia
Miguel Jiménez, miguel@uvic.ca, University of Victoria, Victoria, British Columbia, Canada

ABSTRACT

Towards Continuous Monitoring in Personalized Healthcare through Digital Twins

Luis F. Rivera, rivera@uvic.ca, University of Victoria, Victoria, British Columbia, Canada
Miguel Jiménez, miguel@uvic.ca, University of Victoria, Victoria, British Columbia, Canada
Prashanti Angara, pangara@uvic.ca, University of Victoria, Victoria, British Columbia, Canada
Norha M. Villegas, nvillega@icesi.edu.co, Universidad Icesi, Cali, Valle del Cauca, Colombia, University of Victoria, Victoria, British Columbia, Canada
Gabriel Tamura, gtamura@icesi.edu.co, Universidad Icesi, Cali, Valle del Cauca, Colombia, University of Victoria, Victoria, British Columbia, Canada
Hausi A. Müller, hausi@uvic.ca, University of Victoria, Victoria, British Columbia, Canada

ABSTRACT

Continuous and effective monitoring of chronic diseases and their associated treatments might have a decisive impact on reducing risks and improving life quality of patients. This, however, demands new and innovative methods for engineering systems that support the required capabilities. Research on the application of the novel concept of Digital Twin (DT) in healthcare might provide the means to revolutionize traditional medical practices. A DT comprises a set of virtual representations of both the structural elements and dynamics of any physical asset (e.g., a patient) throughout its lifecycle. In the healthcare domain, it might represent a significant step forward towards tightening and improving the interactions between systems, caregivers and patients. Moreover, integrating data-driven methods (e.g., Machine Learning) and DT could serve as a noteworthy mechanism to not only track patients' health continuously, but also to evaluate the application and evolution of medical treatments virtually. In this paper, we describe our vision for the application of DT in precision medicine. Our contributions are twofold. First, we describe our initial ideas for a reference model that leverages DT capabilities and research advances in self-adaptive systems and autonomic computing to engineer smart and flexible software systems in healthcare. We expect these systems to alleviate complexity and assist in the planning and decision-making processes when applying medical treatments to patients by healthcare professionals. Then, we elaborate on the definition of internal structures for DT to support precision medicine techniques in the context of continuous monitoring and personalized data-driven medical treatments.

CCS CONCEPTS

• Social and professional topics → Medical technologies; • Applied computing → Consumer health; Health care information systems.

KEYWORDS

Digital twin, models at run-time, adaptive context monitoring, self-adaptive systems, autonomic computing, healthcare, mhealth, precision medicine, personalized healthcare, software as a medical device

ACM Reference Format:
Luis F. Rivera, Miguel Jiménez, Prashanti Angara, Norha M. Villegas, Gabriel Tamura, and Hausi A. Müller. 2019. Towards Continuous Monitoring in Personalized Healthcare through Digital Twins. In *CASCON 2019: 29th Annual International Conference on Computer Science and Software Engineering, November, 2019, Markham, Ontario, Canada.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

The recent information explosion together with the instrumentation of artificial intelligence capabilities in modern software systems are enabling new research opportunities and innovation in the healthcare domain [16, 37, 41]. New emerging and interrelated paradigms such as *Informatics & Data-Driven Medicine*,[1] *eHealth*[2] and *mHealth*,[3] *Software as a Medical Device* (SaMD),[4] and *Digital Health*[5] are flourishing and gaining acceptance among both healthcare professionals and patients [11]. This confluence of new available medical data, sensing and data processing technologies, and the design and implementation of smart systems is expected to allow unprecedented levels of precision and personalization in healthcare and medicine. On the one hand, the successful implementation of real-time analysis through continuous monitoring of patients will improve healthcare practices that directly impact

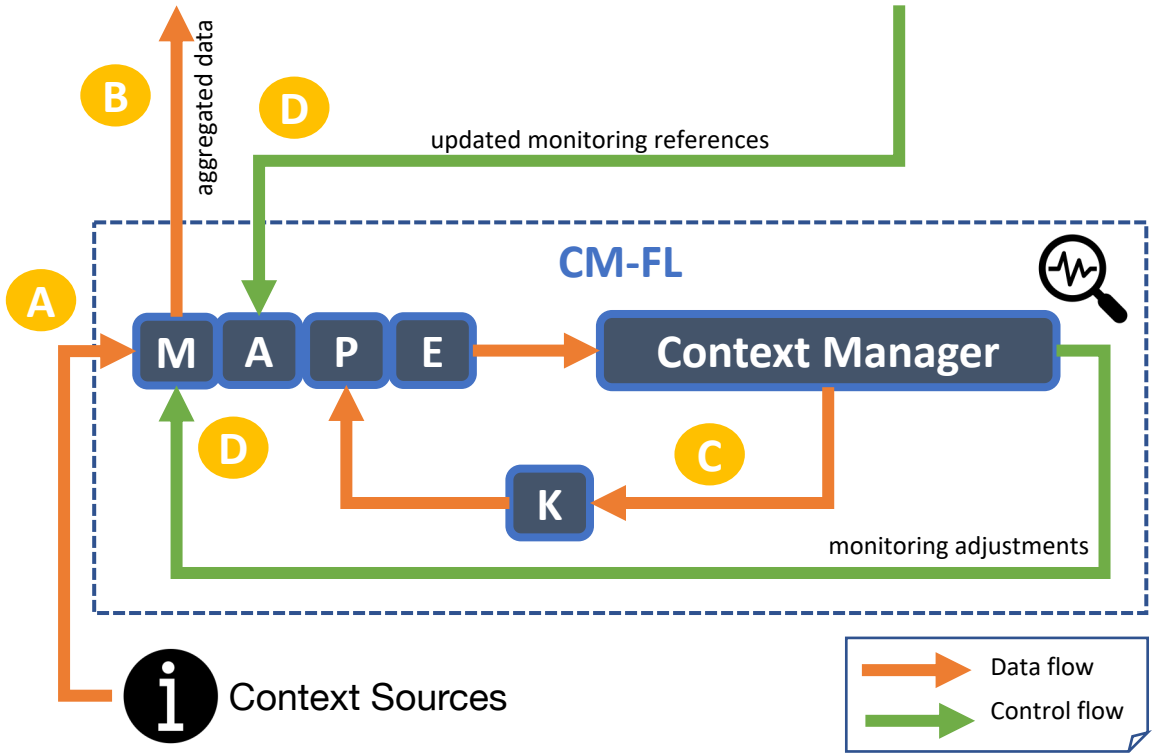
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
CASCON 19, November 2019, Toronto, Ontario, Canada
© 2019 Copyright held by the owner/author(s).

[1] http://med.stanford.edu/iddm/overview.html
[2] https://www.who.int/ehealth/en/
[3] https://www.who.int/goe/publications/goe_mhealth_web.pdf
[4] https://www.fda.gov/medical-devices/digital-health/software-medical-device-samd
[5] https://www.who.int/behealthy/digital-health

GEMINIS – The Context Management Feedback Loop (CM-FL)

Core responsibilities:

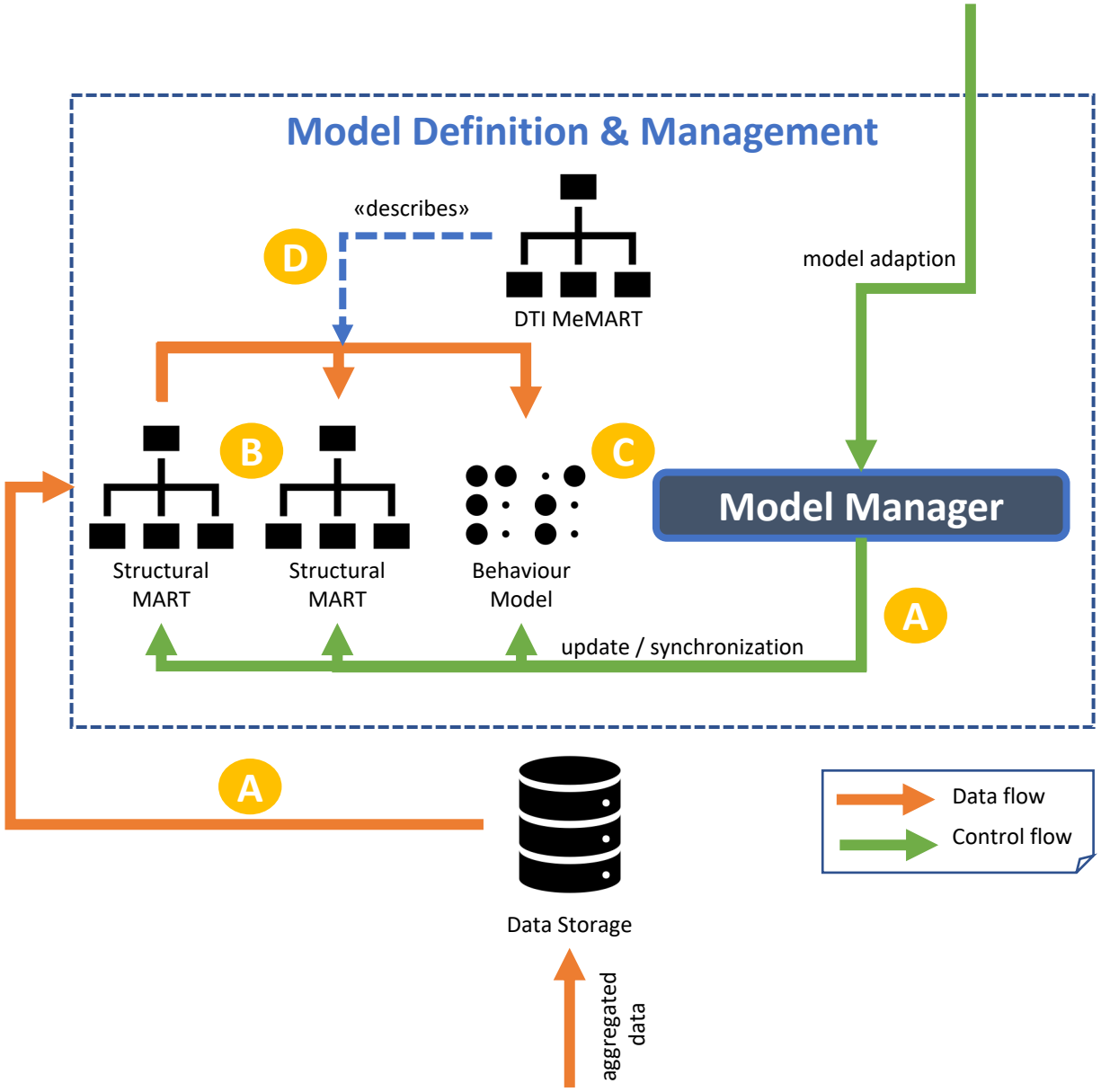
- **A** Acquisition and management of context data
- **B** Data Aggregation
- **C** Detect and consolidate context symptoms
- **D** Perform adaptations of monitoring schemes



GEMINIS – Model Definition and Management

Core responsibilities:

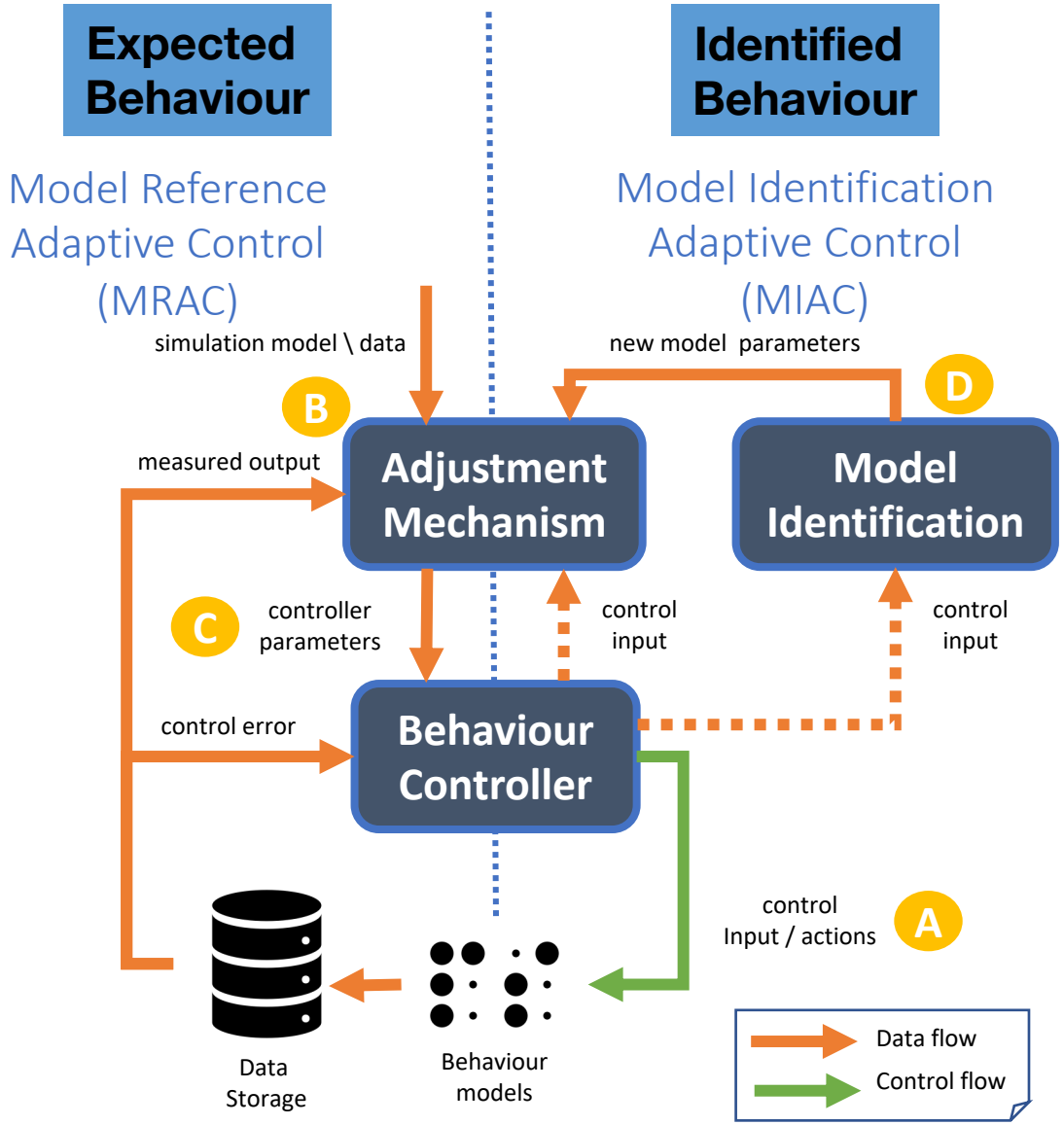
- **A** Operation and synchronization of models
- **B** Real-time status of real twins' relevant characteristics
- **C** Behavioural descriptions and predictions through AI models
- **D** Definition and maintenance of model interrelationships



GEMINIS – The MRAC\MIAC Feedback Loop (MM-FL)

Core responsibilities:

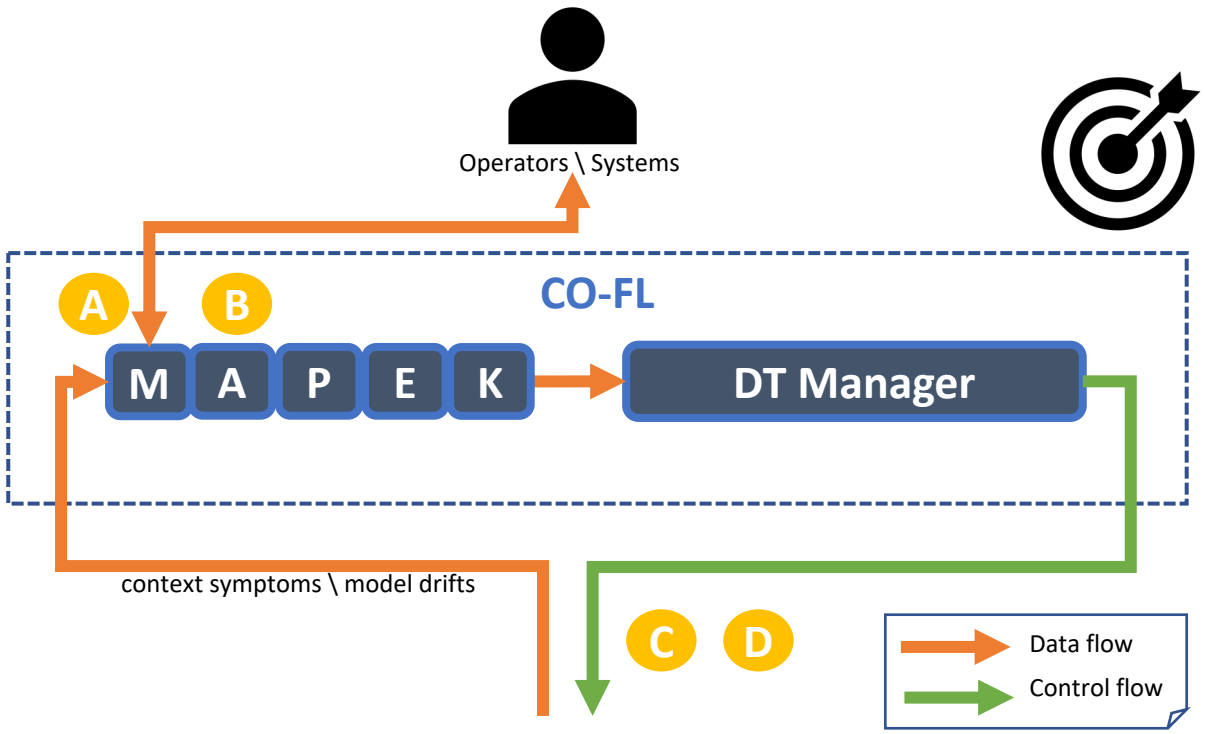
- **A** Continuous evolution of behaviour models
- **B** Comparison between expected and actual real behaviour
- **C** Detection of deviations from expected behaviour and appropriate adjustments (MRAC)
- **D** Tuning based on model identification (MIAC)



GEMINIS – The MRAC\MIAC Feedback Loop (MM-FL)

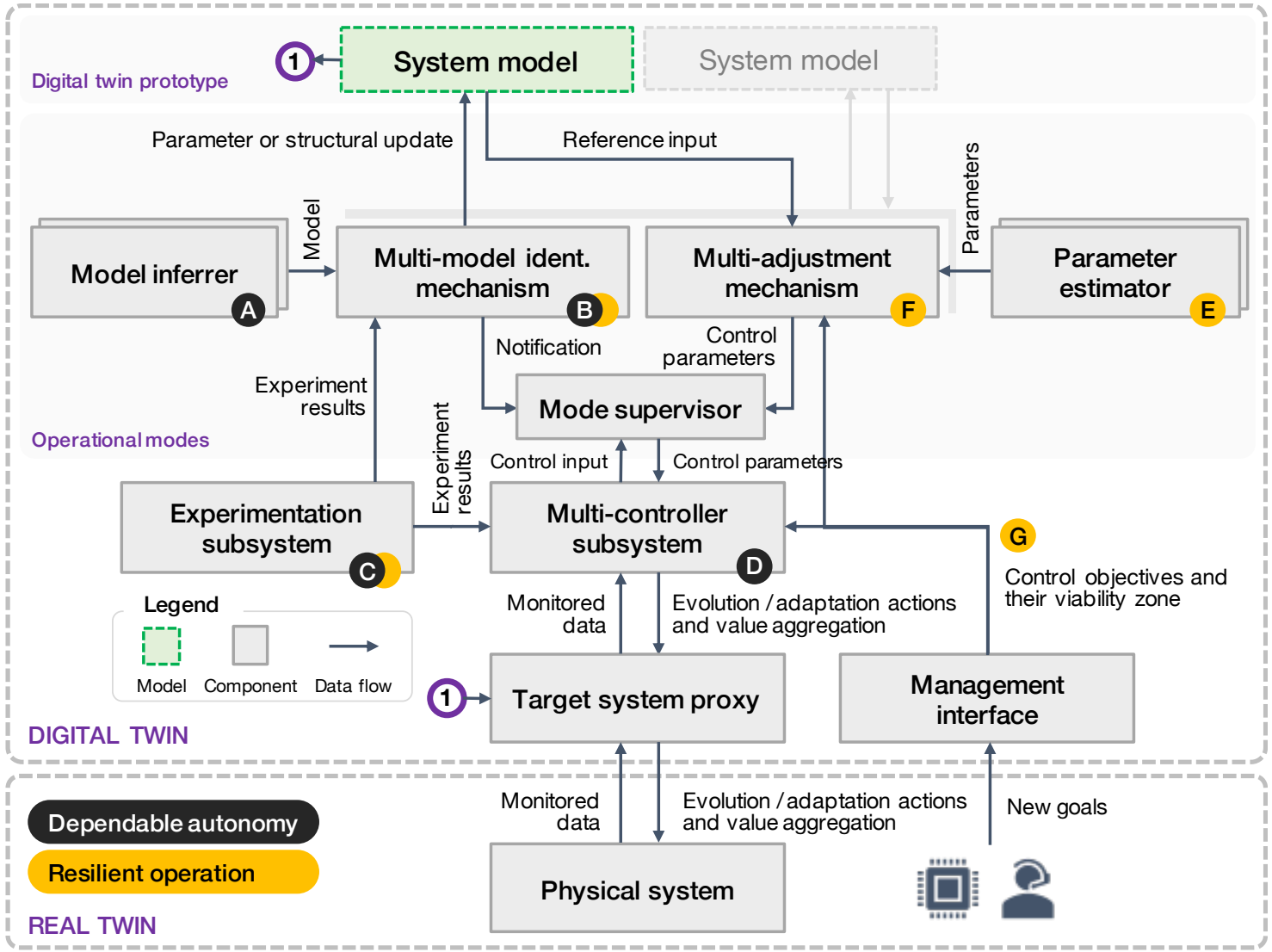
Core responsibilities:

- **A** Satisfaction of high-level goals
- **B** Analysis of critical conditions of real twins or their environments
- **C** Model evolution orchestration
- **D** Digital Twin instantiation



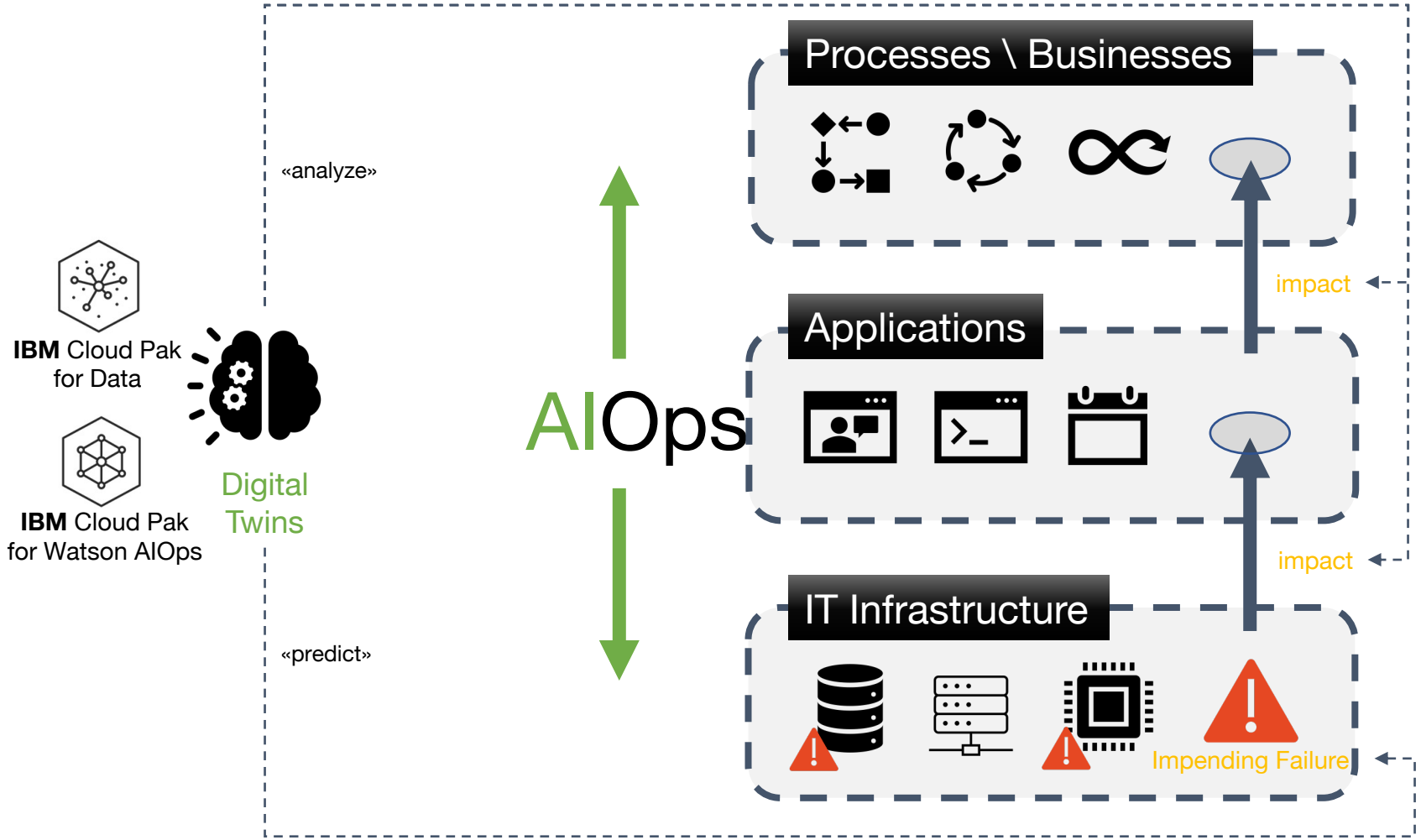
Adaptation and evolution synergy in DTs

- Reference architecture that emphasizes the duality between adaptation and evolution concretely
- Active knowledge reification
- Integrates the concept of evolution with run-time V&V, control objectives, viability zones, run-time models



DTs & AIOps

Shift-Right Approach



L.F. Rivera et al. **The Forging of Autonomic and Cooperating Digital Twins**. Internet Computing. 2021
L.F. Rivera, M. Jiménez et al. **Designing Run-time Evolution for Dependable and Resilient Cyber-Physical Systems using Digital Twins.** JIDPS (Submitted). 2021
L.F. Rivera et al. ***On the Engineering of IoT-Intensive Digital Twin Software Systems***. SERP4IoT (ICSE-Workshops). 2020.

DTs & AIOps

Shift-Left Approach

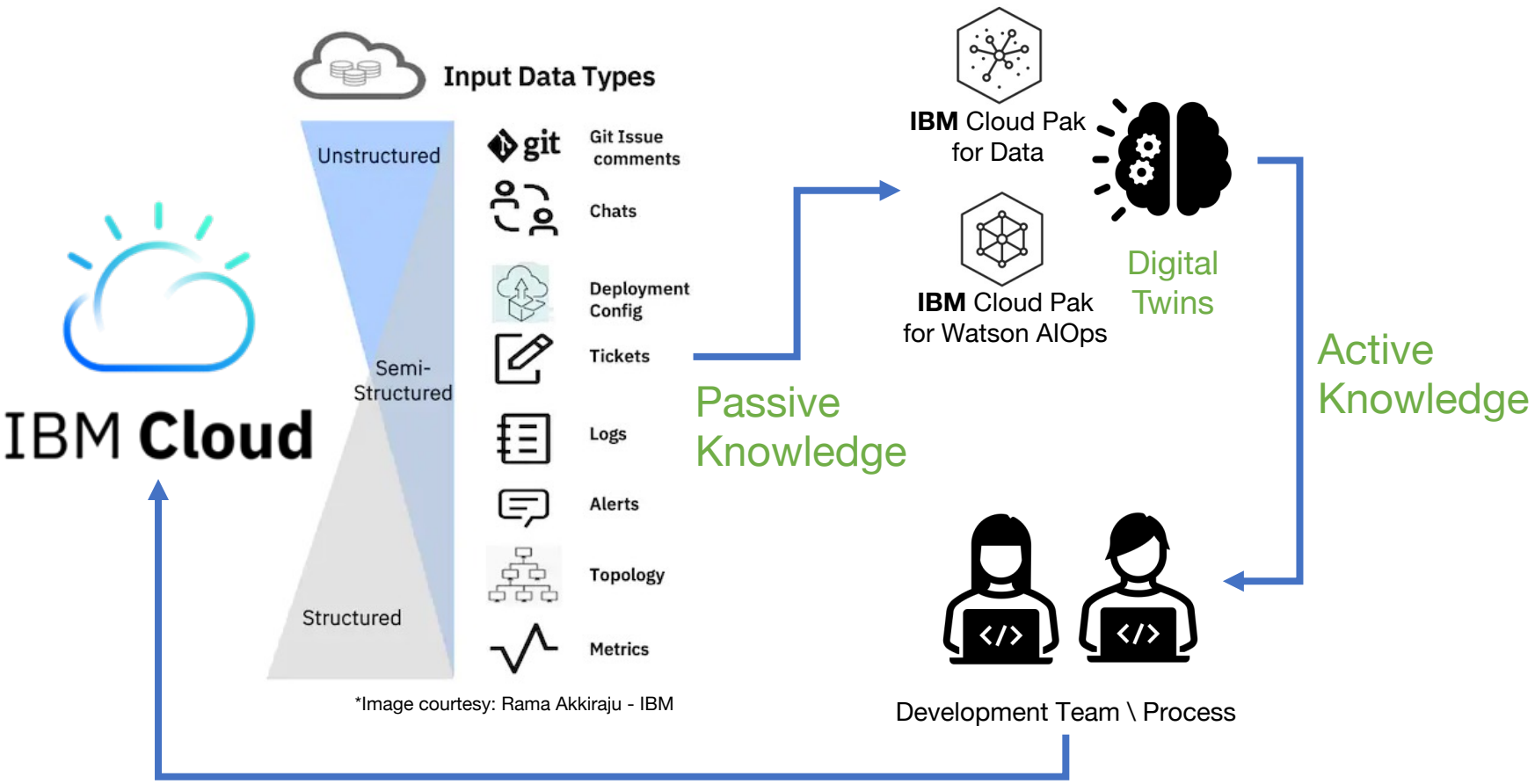


*Image courtesy: Rama Akkiraju - IBM

R. Akkiraju. **Watson AIOps: Bringing AI to IT Operations Management**. IBM. 2020

L.F. Rivera, M. Jiménez et al. **Designing Run-time Evolution for Dependable and Resilient Cyber-Physical Systems using Digital Twins.** JIDPS (Submitted). 2021

Take home messages

1. The innate evolutive nature of real-world environments translates to modern software systems
2. Self-adaptation and self-evolution mechanisms are key enablers of smart software systems such as DTs

1. There are remaining discontinuities in the continuous delivery process
2. The execution environment affects the software evolution. We must make this explicit!
3. We are working toward a run-time evolution process. Imagine Soft. Eng. @ run-time

Key takeaway:

- Autonomic and cooperative next-generation systems will conduct engineering processes at run-time to self-evolve

Thank you!

We would like to thank the following organizations for making this research possible:

University of Victoria

UNIVERSIDAD ICESI

NSERC CRSNG

DITA

CAS

Questions?